UiO : University of Oslo

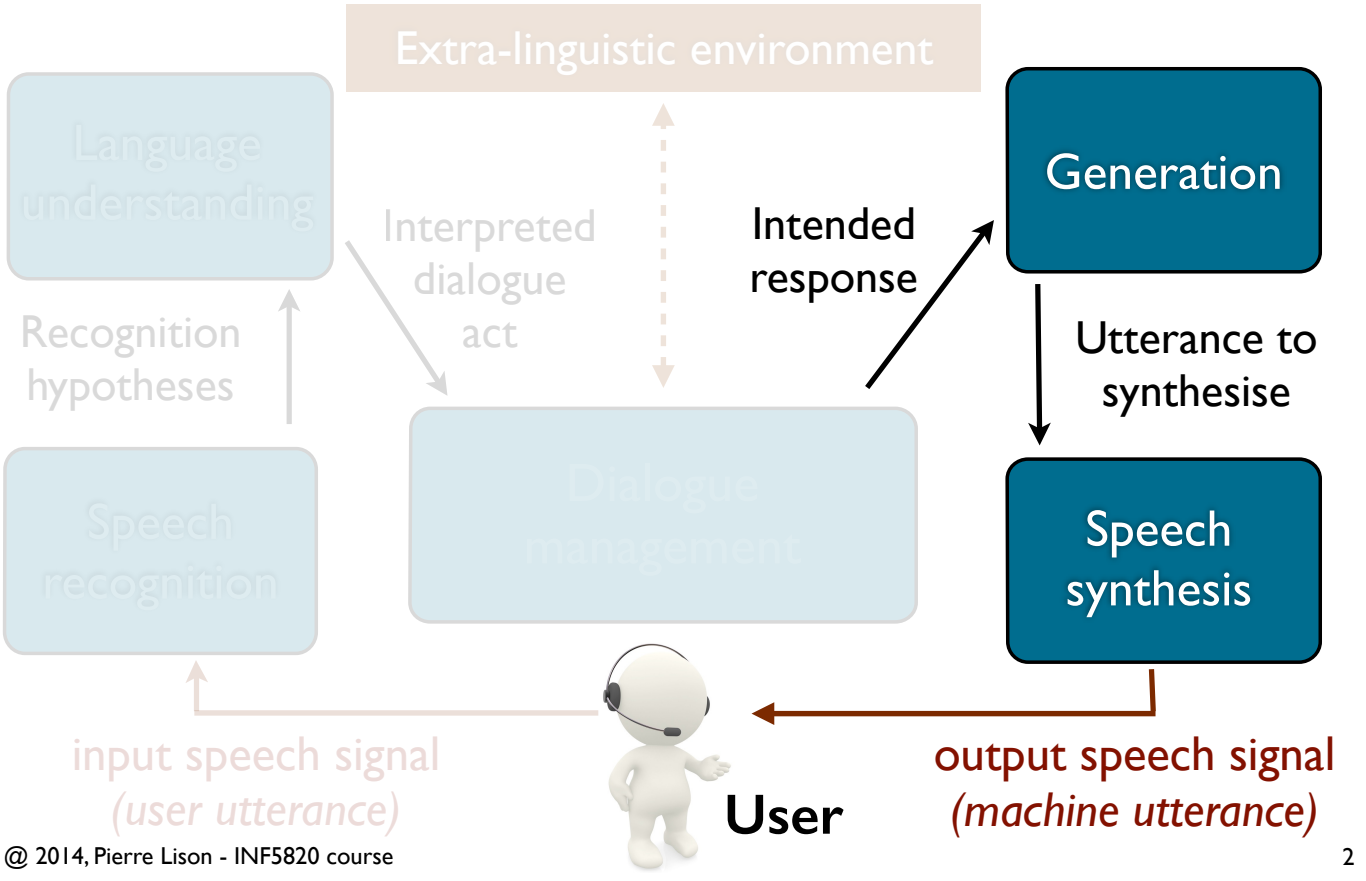
INF5820: Generation & synthesis

Pierre Lison,
Language Technology Group (LTG)
Department of Informatics

Fall 2014

UiO : University of Oslo

Generation and synthesis



Outline

- Natural language generation
- Speech synthesis
- Summary
- Final wrap-up

Outline

- **Natural language generation**
  - **Shallow, deep and statistical approaches**
  - **Generating referring expressions**
- Speech synthesis
- Summary
- Final wrap-up

Natural language generation

- The goal of NLG is to convert a high-level communicative goal into a concrete utterance
- As for NLU, a wide range of methods exists,with varying degrees of complexity:
  - *Shallow approaches* based on canned utterances
  - *Deep approaches* based on generic grammatical resources and reasoning patterns
  - *Statistical approaches* based on observed data

Shallow NLG

- Shallow approaches to NLG
  - the system designer manually maps the communicative goals a_m to specific handcrafted utterances u_m
  - The utterances might contain slots to be filled

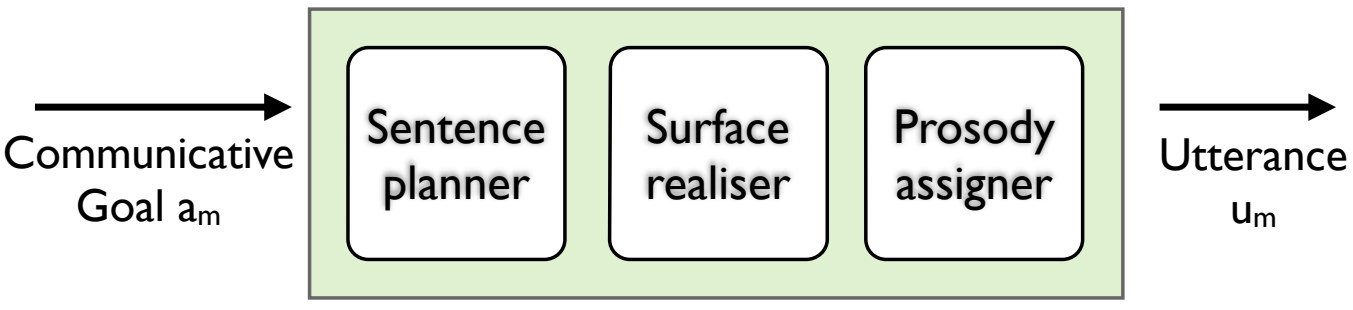
Goal a_m	Utterance u_m
AskRepeat	«Sorry, could you please repeat?»
Assert(cost(ticket, price))	«This ticket will cost you {price} USD»
Ask(departure)	«Please state where you are flying from» «Where are you departing from?»

Shallow NLG

- Shallow approaches are by far the most popular in commercial systems
  - Limited effort: there are rarely more than a few hundreds prompts for a given system
  - Gives the designer full control over the system behaviour (important for quality assurance)
- One can introduce some variation by randomly selecting the utterance from a set of possible candidates

Deep NLG

- Shallow approaches rely on the detailed specification of every possible utterance
- A good part of this process is domain-independent and could be automatised



Deep NLG pipeline

Deep NLG

- Pipeline of modules:
  - **Sentence planning**: selection of *abstract linguistic items* (lexemes, semantic structure) to achieve the communicative goal.
  - **Surface realisation**: construction of a *surface utterance* based on the abstract items and language-specific constraints (word order, morphology, etc.)
  - **Prosody assignment**: determination of the utterance's *prosodic structure* based on information structure (e.g. what is in focus, what is given vs. what is new)

Often conflated in many practical generation systems

Sentence planning

- How to perform sentence planning?
  - Recall Grice's **cooperative principle**, and in particular the Maxim of Quantity: *say exactly as much as is necessary for your contribution*
  - The goal is therefore to find the best way to convey the system's intention, in the fewest possible words
  - ... but while remaining clear and unambiguous!
  - The communicative goal must sometimes be split in several separate utterances

Surface realisation

- Given a high-level semantics of the utterance provided by the sentence planner, one can *realise* it in a concrete utterance
  - This is the inverse operation as classical parsing!
- Some grammatical formalisms are «bidirectional» or reversible, i.e. they can be used for both parsing and generation
  - HPSG or CCG grammars are reversible (at least can be made reversible, given some grammar engineering)

Prosodic assignment

- Information structure:
  - *theme*: part of an utterance which is talked about (given)
  - *rheme*: what is said about the theme (new)
- Linguistic realisation of this structure in word order, syntax and intonation

Q: I know the AMERICAN amplifier produces MUDDY treble,
Q1: but WHAT does the BRITISH amplifier produce?
A1: (The BRITISH amplifier produces)$_{th}$ (CLEAN treble.)$_{rh}$
L+H* LH% H* LL$
Q2: but WHAT produces CLEAN treble?
A2: (The BRITISH amplifier)$_{rh}$ (produces CLEAN treble.)$_{th}$
H* LL% L+H* LH$

[S. Prevost (1996) «A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation», PhD thesis]

Statistical generation

- Deep approaches to generation can be «brittle»:
  - Requires fine-grained *grammatical resources*
  - Need to *rank* large numbers of alternative utterances produced for a communicative goal
  - ... according to which *quality measures*?
  - *User adaptation* is difficult
- An alternative is to rely on statistical techniques to *learn from data* the best way to express a given communicative goal

Statistical generation

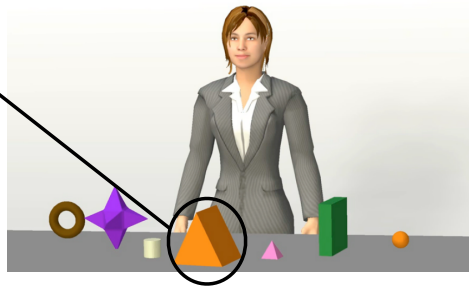
- Learn from what kind of data?
- Supervised learning: learn from annotated examples
  - Requires "gold standard" examples of system utterances for a given communicative goal and context
  - The examples can be collected from e.g. Wizard-of-Oz interactions
- Reinforcement learning: learn from trial-and-error
  - Let the system explores various ways to generate a given goal
  - Update the generation strategy based on the received feedback

[Verena Rieser, Oliver Lemon (2010), «Natural Language Generation as Planning under Uncertainty for Spoken Dialogue Systems». *Empirical Methods in Natural Language Generation*]

Generation of referring expressions

- *Generating referring expressions* (GRE) is an interesting subproblem of NLG
- Objective: given a reference to an object/entity in the context, find the best referring expression for it!

Let's say we want to talk about this object



The object?

The triangular object?

The orange triangular object that is to the right of the pink pyramid and to the left of the white cylinder?

Generation of referring expressions

- GRE typically searches for the *minimal distinguishing expression* for the target
- A distinguishing expression matches the target, but none of the *distractors* (other salient objects in the context)

Target

Distractors

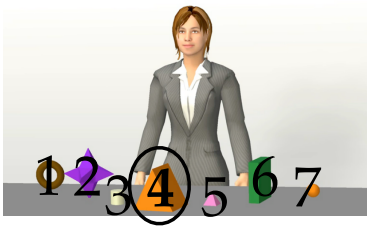
Generation of referring expressions

- Dale and Reiter's Incremental Algorithm:
  1. order the properties P by preference
  2. Iterate through ordered list of properties P
  3. add attribute to description being constructed if it rules out any remaining distractors
  4. terminate when a distinguishing description has been constructed (or no more properties)

[Robert Dale and Ehud Reiter (1995), «Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions». *Cognitive Science*]

Incremental algorithm: example

- Assume three properties: Shape, Colour and Size, with Shape > Colour > Size
- We want to talk about object 4



Step	Current expression	Remaining distractors
We analyse the Shape property. Object 4 has Shape=triangular	The object	{1,2,3,5,6,7}
Adding the property Shape=triangular removes distractors {1,2,3,6,7}	The triangular object	{5}
We analyse the Colour property. Object 4 has Colour=orange	The triangular object	
Adding the property Colour=orange remove the distractor 5	The orange triangular object	∅
Found distinguishing expression!	The orange triangular object	∅

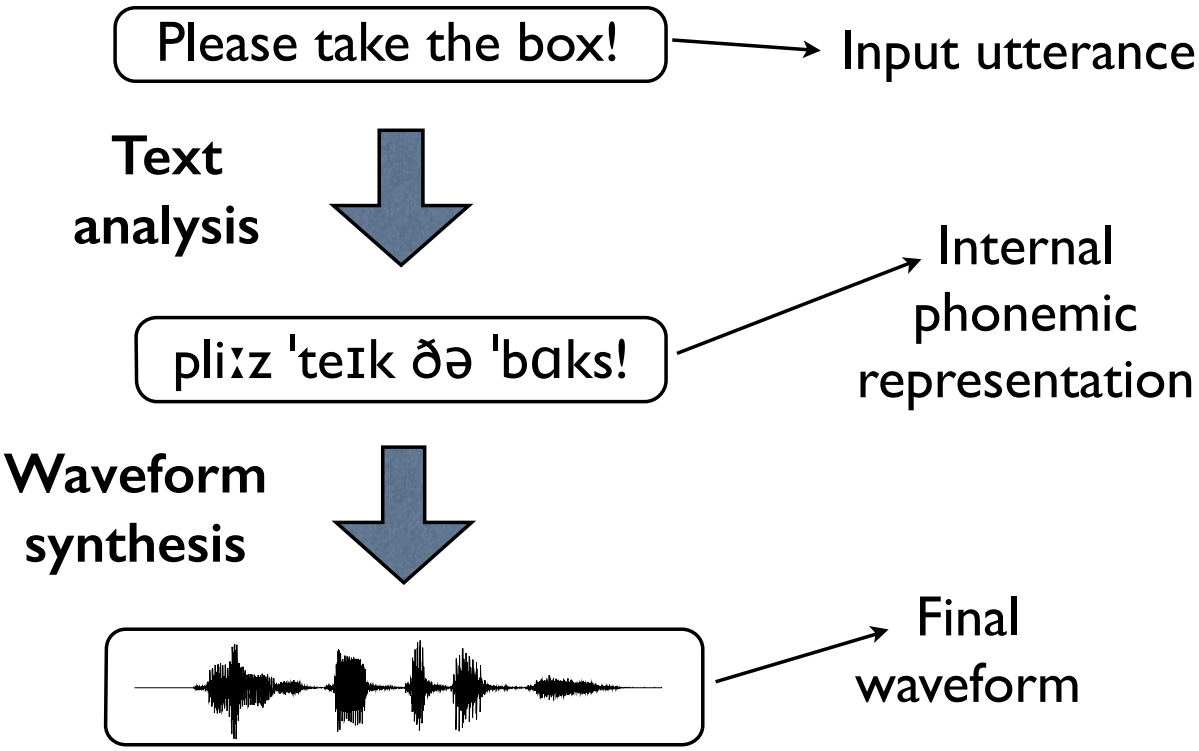
Outline

- Natural language generation
- **Speech synthesis:**
  - **Text analysis**
  - **Waveform synthesis**
- Summary
- Final wrap-up

Speech synthesis

- The last component of our architecture is the speech synthesiser (or «text-to-speech», TTS)
- The TTS module converts a concrete utterance into a speech waveform
- This mapping is performed in two steps:
  1. Conversion of *input utterance* into a *phonemic representation* (**text analysis**)
  2. Conversion of *phonemic representation* into the *waveform* (**waveform synthesis**)

Speech synthesis



Text analysis in TTS

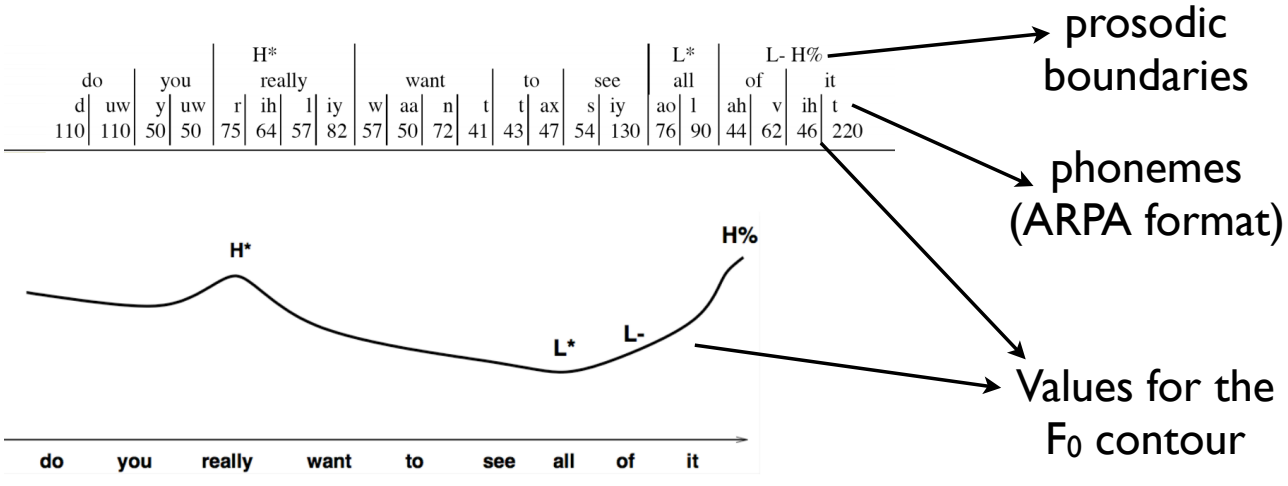
- How do we produce the phonemic representation?
  1. *Text normalisation* (abbreviations, numbers, etc.)
  2. *Phonetic analysis*, based on a pronunciation dictionary and a grapheme-to-phoneme (g2p) converter
  3. *Prosodic analysis* to determine e.g. prosodic phrases, pitch accents, and overall tune

Prosodic analysis

- Utterances can be structured in *intonational phrases*
  - Correlated, but not identical to syntactic phrases!
  - These phrases can be extracted based on features such as punctuation, presence of function words etc.
- Words can be more or less prosodically *prominent*
  - E.g. emphatic accents, pitch accents, unaccented, reduced
- Finally, utterances are also characterised by their global *tune* (rise and fall of F_0 over time)

Phonemic representation

At the end of the text analysis (normalisation + phonemic and prosodic analysis), we end up with an internal phonemic representation of our utterance



Waveform synthesis

- Once we have a phonemic representation, we need to convert it into a waveform
- Two families of methods:
  - **Concatenative synthesis**: glue together pre-recorded units of speech (taken from a speech corpus)
  - **Formant & articulatory synthesis**: generate sounds using acoustic models of the vocal tract

Waveform synthesis

	Pros	Cons
Concatenative synthesis	• More natural-sounding & intelligible speech • Easier modelling, limited signal processing	• Requires a speech corpus • Limited flexibility
Formant and articulatory synthesis	• Explicit model of speech production • Many parameters can be tweaked	• Robotic sounds • Complex modelling and signal processing

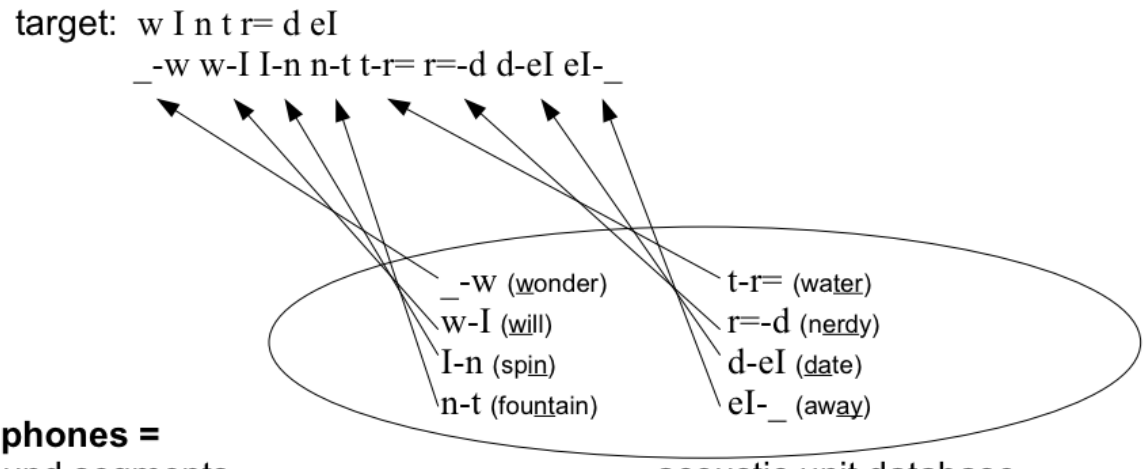
Concatenative synthesis

- Concatenative synthesis:
  - We record and store various units of speech in a database
  - When synthesising a sound, we search the appropriate segments in this database
  - We then «glue» them together to produce a fluent sound

Concatenative synthesis

- Concatenative methods differ by the kind of «units of speech» they are using
  - **Diphone synthesis**: phone-like units going from the middle of one phone to the middle of the next one
  - **Unit selection**: units of different sizes, can be much larger than a diphone
- Most commercial TTS systems deployed today are based on unit selection

Diphone synthesis



Diphones =
sound segments
from the middle of one phone
to the middle of the next phone

acoustic unit database
units = **diphone segments**
recorded in carrier words
(flat intonation)

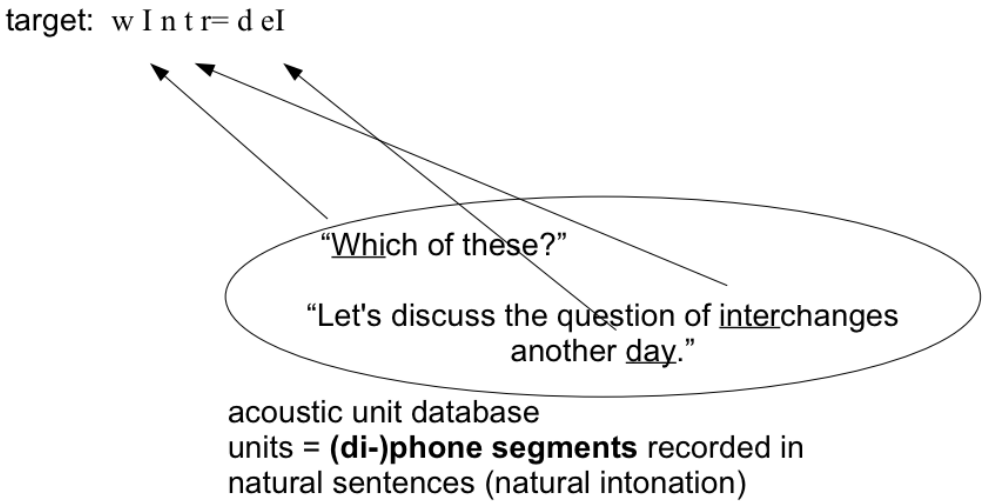
[diagram borrowed from M. Schröder]

Diphone synthesis

- For diphone synthesis, the acoustic database consists of recorded diphones
  - Usually embedded in carrier phrases
  - Must be carefully segmented, labelled, pitch-marked, etc.
- After concatenation, the sound must be adjusted to meet the desired prosody
  - This signal processing might distort the speech sound!
  - Limited account of pronunciation variation (only coarticulation due to neighbouring phone)

Unit selection synthesis

In unit selection synthesis, the «units of speech» come from a segmented corpus of natural speech



[diagram borrowed from M. Schröder]

Unit selection synthesis

- How do we search for the best units matching our phonemic specifications?
  - Search for a unit that matches as closely as possible our requirements (F_0, stress level, etc.) for the unit
  - ... and that concatenates smoothly with its neighbours
- Given a specification s_t, we search for the unit u_t that minimises two costs:
  - *Target cost* $T(u_t, s_t)$: how well the specification matches u_t
  - *Join cost* $J(u_t, u_{t+1})$: how well u_t joins with its neighbour u_{t+1}

Unit selection synthesis

- Assume that we are given an internal phonemic representation $S=\{s_1,s_2,...s_n\}$
- We want to find the best sequence of speech units for S
- In other words, we search for the unit sequence $\hat{U}=\{u_1,u_2,...u_n\}$ such that:

$$\hat{U} = \underset{U}{\operatorname{argmin}} \sum_{t=1}^{n} T(s_t, u_t) + \sum_{t=1}^{n-1} J(u_t, u_{t+1})$$

Target cost between specification s_t and unit u_t

Join cost between unit u_t and unit u_{t+1}

Unit selection synthesis

- Unit selection can produce high-quality sounds
  - Depending on the corpus size and quality, of course
- But it's rather inflexible: difficult to modulate the prosody of the speech sound
  - How can we e.g. change the sound's emotional content?
  - Alternative: annotate the speech corpus with fine-grained informations, and use these in the selection
  - But requires a much larger corpus!

Outline

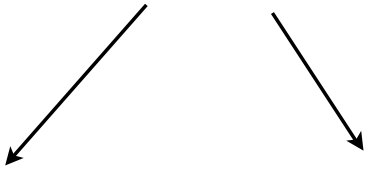
- Natural language generation
- Speech synthesis
- **Summary**
- Final wrap-up

Summary

- Approaches to natural language generation:
  - Shallow methods based on canned utterances
  - Deep methods based on grammars and logical reasoning
  - Statistical methods based on patterns learned from data
- Special case: *generating referring expressions* (GRE):
  - Find the best linguistic expression that identifies a given entity
  - Need to find an expression which is both *distinguishing* (matches the target object, but no other object) and *minimal*

Summary

- Speech synthesis task:
  - *First step*: convert the utterance into an internal phonemic representation, together with a prosodic structure
  - *Second step*: convert this representation into a waveform



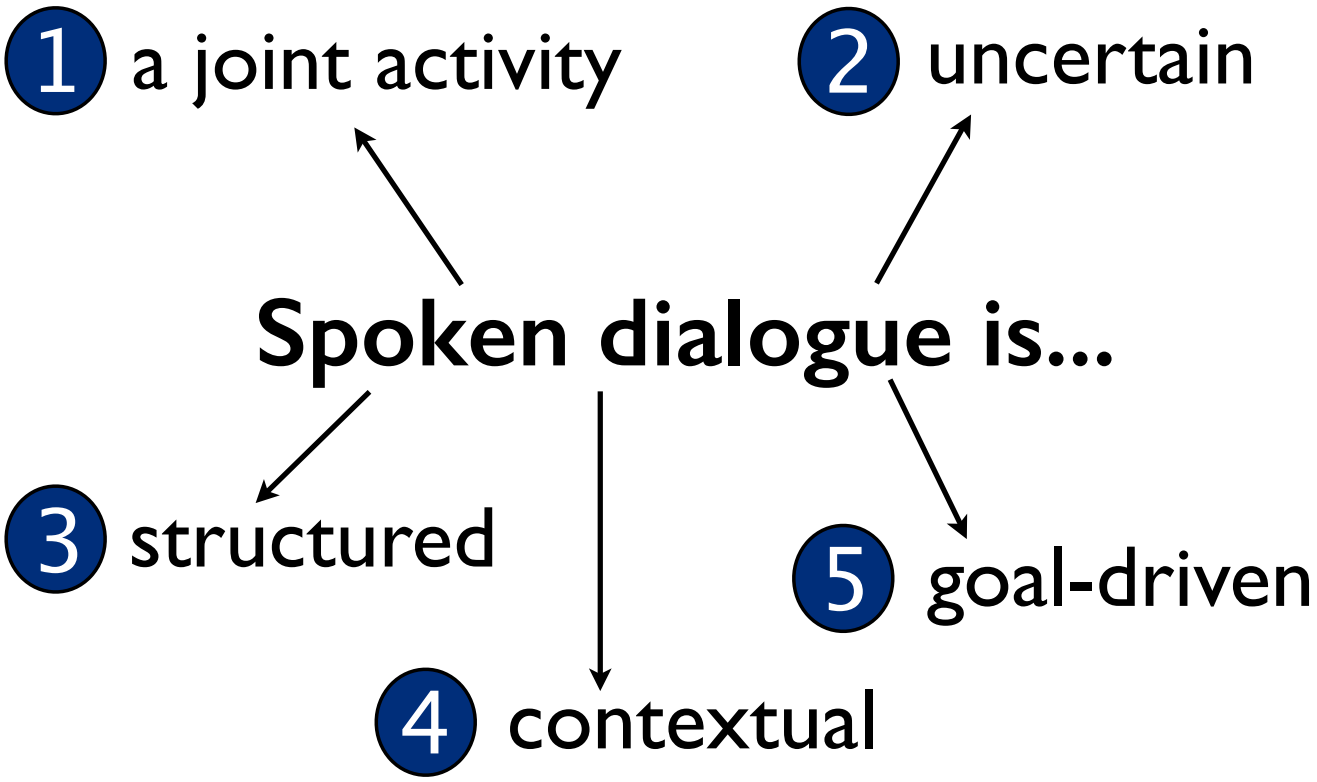
Concatenative synthesis (diphone, unit selection): reuse pre-recorded units from an acoustic database

Formant & articulatory synthesis: relies on an explicit acoustic model of the vocal tract

Outline

- Natural language generation
- Speech synthesis
- Summary
- **Final wrap-up**

Wrap-up



Dialogue is a joint activity

The dialogue participants take *turns* and perform *dialogue acts*

They cooperate to achieve mutual understanding (*common ground*)

They *interpret* each others utterances cooperatively

They naturally *align* their way of speaking

Dialogue is uncertain

- **Uncertainty** is everywhere in dialogue:
  - Error-prone speech recognition
  - Ambiguities at multiple levels
  - Unpredictable environments and user behaviors
  - Dialogue context often only partially observable

That's why *probabilistic modelling* is a key element in the design of robust dialogue systems

Dialogue is structured

- Dialogue is structured in many (overlapping) levels:
  - *Syntactic* and *prosodic* phrases
  - *Semantic relations* within an utterance
  - *Pragmatic relations* between utterances
  - *References* to external entities, persons, places, events
  - *Attentional* structure (which entities are in focus)
  - *Intentional* structure (which high-level goals is each participant trying to achieve in the dialogue)

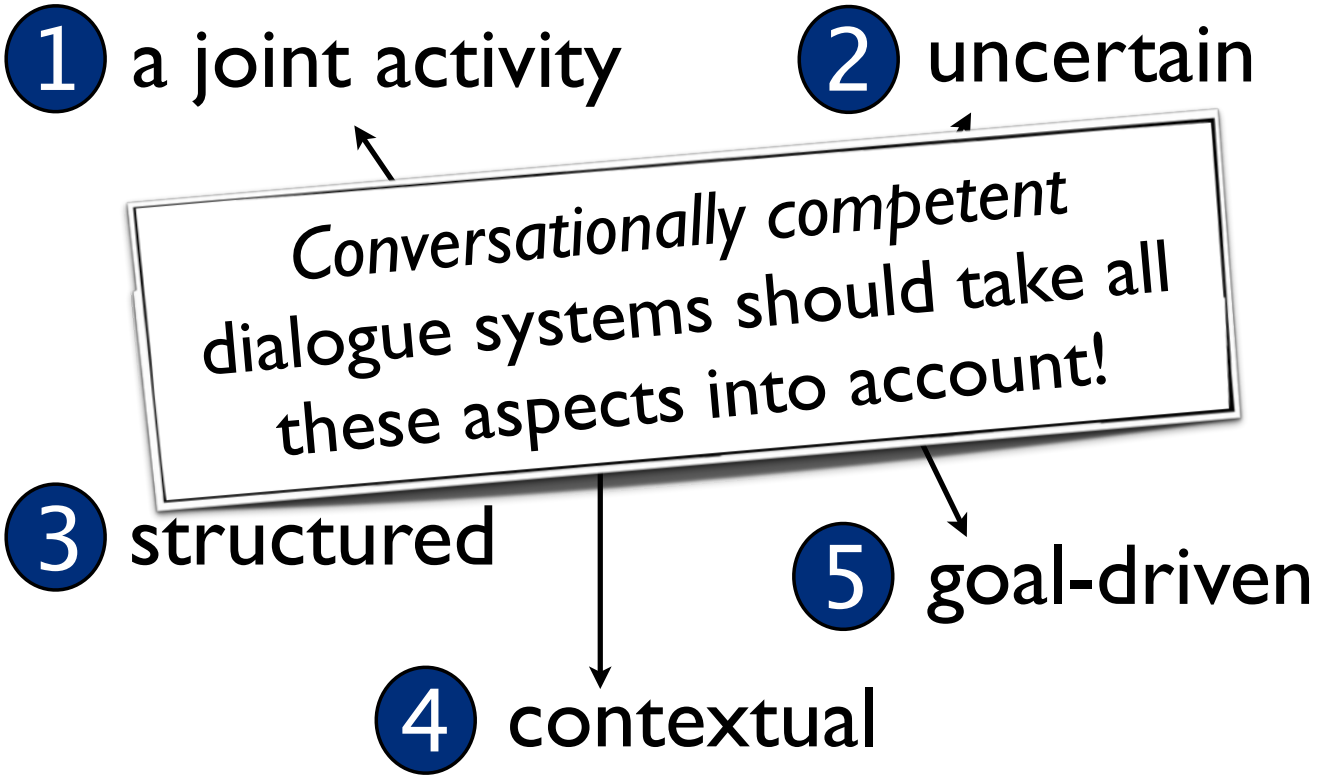
Dialogue is contextual

- **Context** is crucial for dialogue processing:
  - *Pronunciation* varies depending on the context
  - Most utterances only make sense in a *situation*
  - Omnipresence of *deictics* in dialogue
  - Virtually all dialogue processing tasks must adapt their output based on *contextual factors*
  - Need to continuously track the current dialogue state!

Dialogue is goal-driven

- We communicate to **do things** in the world
  - *Dialogue acts* guided by *intentions* and provoking *effects*
  - *Verbal* and *non-verbal* actions are intertwined
  - Dialogue participants have multiple competing goals to fulfill, leading to a problem of *utility maximisation*
  - To this end, they must *plan* their actions over time

Wrap-up



Practical details

- To prepare for the exam:
  - List of relevant sections from Martin & Jurafsky's book
  - Exam questions from 2012 and 2013 (make-up exam)
  - Exam-like exercises for both MT and dialogue
  - ... all available on the course website
- Contact me by email if you have questions
- Good luck!